Documentation

1. Data Cleaning

- Imputed missing values (mean, linear interpolation)
- Removed duplicate entries
- Converted month columns to numeric
- Standardized city name formats

Part 1: Data preperation and cleaning

Generate + Code + Markdown

```python
df = pd.read_csv("data/air_quality_dataset.csv")
print(df.info()) #check info
```

[6] ✓ 0.0s Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2164 entries, 0 to 2163
Data columns (total 16 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Rank     2164 non-null   int64
 1   City     2164 non-null   object
 2   Country  2164 non-null   object
```

Generate + Code + Markdown

```python
## check missing values by columns
df.isnull().sum(axis=0).sort_values(ascending=False)
```

[8] ✓ 0.4s Python

```
Nov        91
Jan        87
Feb        66
May        58
```

270 out of 2164 rows (12.48%) contain at least one missing value. And total amount is 458 missed value

```python
df.iloc[:, -12:] = df.iloc[:, -12:].apply(lambda col: col.fillna(col.mean()), axis=0)#adding mean to missing valuse
sum(df.isnull().sum())#sum of missing values
```

[11] ✓ 0.0s

```
0
```

```python
df.head(20)
```

[12] ✓ 0.0s

	Rank	City	Country	2023	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	1	Begusarai	India	118.9	31.2	235.3	156.8	113.0	109.3	99.0	63.800000	61.8	71.5	61.8	210.5	285.0
1	2	Guwahati	India	105.4	220.2	168.1	129.2	112.2	69.5	51.3	46.600000	60.2	76.7	76.4	126.9	128.0
2	3	Delhi	India	102.1	171.8	114.3	77.4	71.0	67.4	42.9	35.300000	34.8	39.7	106.3	255.1	210.0
3	4	Mullanpur	India	100.4	106.3	123.7	78.1	56.6	53.4	53.9	63.200000	59.7	59.6	110.4	253.0	201.4
4	5	Lahore	Pakistan	99.5	143.2	117.3	73.8	52.9	52.4	46.4	39.800000	42.2	53.8	125.9	251.0	197.5
5	6	New Delhi	India	92.7	162.6	98.2	67.1	59.0	57.7	40.1	31.700000	35.0	38.0	94.7	234.7	193.8
6	7	Siwan	India	90.6	223.6	167.5	108.3	71.7	59.8	48.9	35.700000	30.4	54.7	48.9	136.3	77.6
7	8	Saharsa	India	89.4	202.0	147.1	108.8	88.8	60.3	43.6	16.700000	24.3	33.6	41.8	115.8	167.8
8	9	Goshaingaon	India	89.3	205.3	117.5	63.8	63.3	60.7	39.2	27.300000	38.3	33.9	81.1	152.0	156.1
9	10	Katihar	India	88.8	224.1	113.3	94.0	74.1	49.9	34.7	17.700000	27.6	33.9	63.9	134.1	180.5

Part 1: Data Cleaning and Initial Exploration Summary

Dataset Description:

The dataset contains information on PM2.5 air pollution levels for 2164 cities across Asia in 2023. It includes columns for the city's rank based on annual PM2.5 average, city and country names, the overall 2023 average, and monthly PM2.5 values from January to December.

Data Cleaning Steps:

The dataset was loaded using pandas.read_csv(). Monthly columns were checked and converted to numeric types using pd.to_numeric(errors='coerce') to ensure consistency and handle any non-numeric values.

Missing values were primarily found in the monthly columns. These rows were dropped before performing modeling using df.dropna(subset=monthly_columns). The dataset was also checked for duplicate entries, and city names were normalized for case consistency (e.g., 'Almaty' vs 'almaty').

Initial Exploration:

The dataset contains 2164 rows and 17 columns, covering over 40 countries in Asia. Descriptive statistics were generated using .describe() to understand the distribution of PM2.5 values. Results showed high variability, with some cities exceeding 100 µg/m³.

City-level analysis was performed for selected locations, such as Almaty, to explore monthly trends. Seasonal changes in pollution levels were evident in many cities.

Initial Exploration:

The dataset contains 2164 rows and 17 columns, covering over 40 countries in Asia. Descriptive statistics were generated using .describe() to understand the distribution of PM2.5 values. Results showed high variability, with some cities exceeding 100 µg/m³.

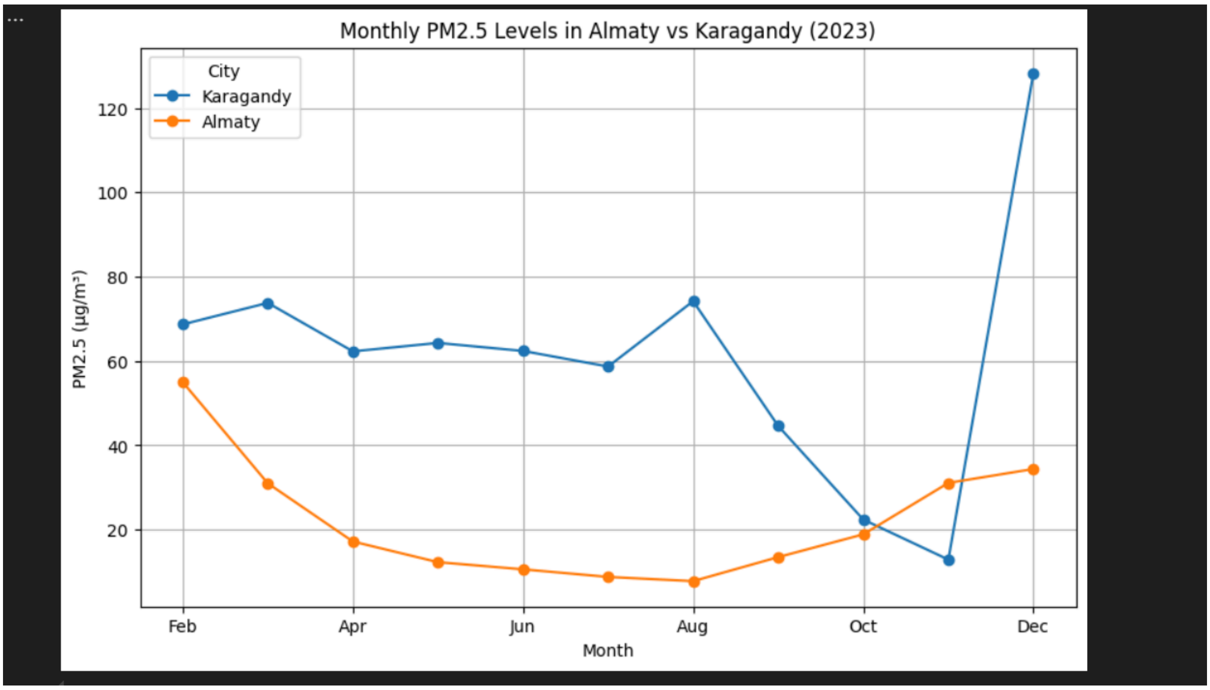
City-level analysis was performed for selected locations, such as Almaty, to explore monthly trends. Seasonal changes in pollution levels were evident in many cities.

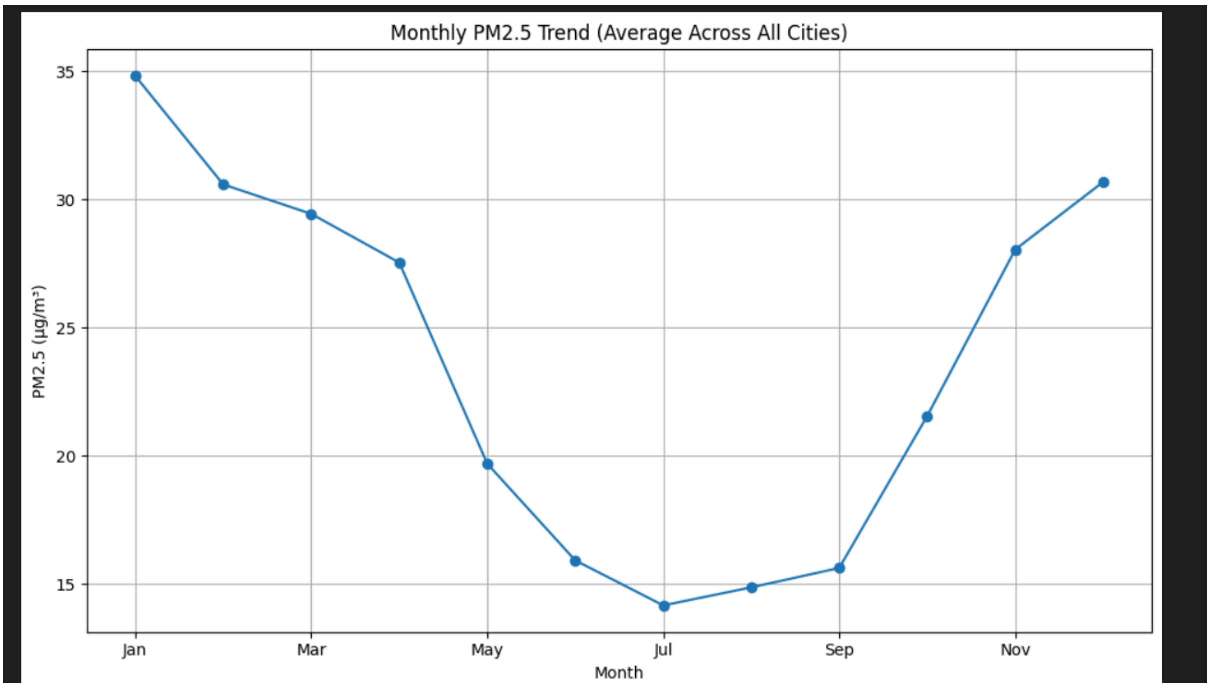
Observations:

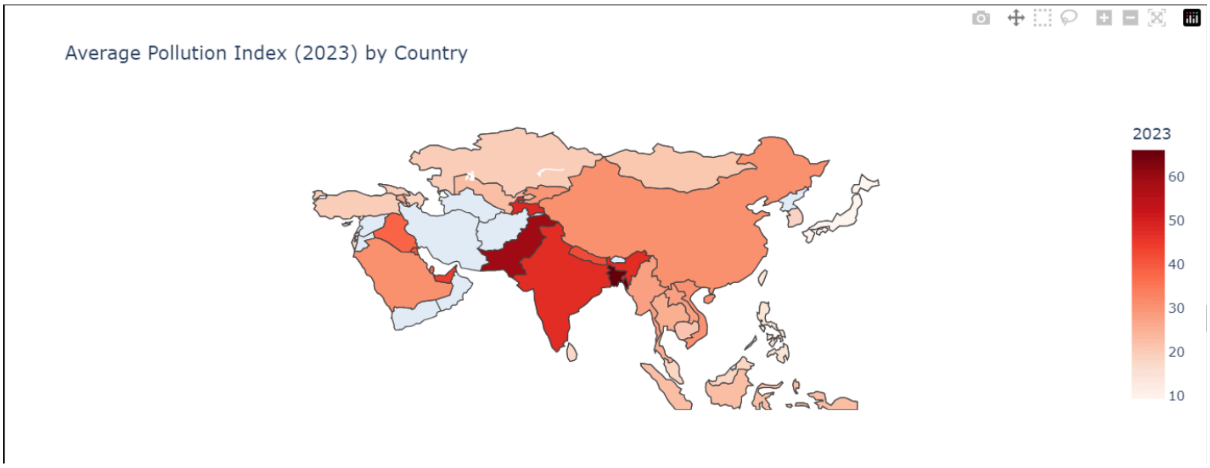
The dataset is largely clean, with some missing data in monthly values. Seasonal variation is significant, with higher pollution during winter months. The distribution of annual PM2.5 levels is right-skewed, with a few cities experiencing extremely high pollution.

2. 📊 Exploratory Data Analysis (EDA)

- Line plots for **monthly trends**
- Bar charts for **city-wise yearly averages**
- Heatmaps showing **seasonal variation** across cities


Monthly PM2.5 Levels in Almaty vs Karagandy (2023)
City
Karagandy
Almaty
120
100
80
60
40
20
PM2.5 (µg/m³)
Feb
Apr
Jun
Aug
Oct
Dec
Month


Monthly PM2.5 Trend (Average Across All Cities)
35
30
25
20
15
PM2.5 (µg/m³)
Jan
Mar
May
Jul
Sep
Nov
Month



EDA Visualization summary

Data Cleaning and Preparation

The dataset containing PM2.5 pollution data for 2164 cities across Asia was loaded and examined. The monthly pollution columns (January to December) were stored in columns 4 to 15 and converted to numeric format to ensure consistency and enable analysis. Rows with missing or invalid numeric values were automatically handled using pd.to_numeric with coercion to NaN. This step was essential to prepare the dataset for accurate aggregation and visualization.

Identifying the Most Polluted Cities

Using the cleaned data, we identified the top 10 most polluted cities in Asia based on their 2023 average PM2.5 levels. These cities were visualized using a horizontal bar plot, clearly showing which urban areas experienced the highest pollution. Additionally, a separate ranking was created specifically for Kazakhstan to highlight the most polluted cities within the country. This allowed us to localize the analysis and provide region-specific insights.

Monthly Trends in Air Pollution

To understand seasonal patterns, we computed the average PM2.5 level across all cities for each month. This monthly trend line revealed a clear seasonal effect, with PM2.5 concentrations peaking during winter months (especially January) and dipping in summer, likely due to factors such as heating emissions, low atmospheric dispersion, and meteorological conditions.

Heatmap Visualization of Monthly Pollution

A heatmap was generated to display monthly PM2.5 levels for the Asia countries. This allowed for a detailed comparison of pollution levels not just annually, but across multiple cities. The visualization highlighted cities with consistently high pollution as well as those that experienced seasonal spikes. Based of this map we can see that India struggles the most with air quality problem and Kazakhstan is in normal state.

3. 🤖 Prediction and Modeling

- **Goal:** Predict 2023 annual PM2.5 using Jan–Dec monthly values
- **Models Used:**
    - Linear Regression

- Random Forest Regressor
- LightGBM Regressor
- CatBoost Regressor
- XGBoost Regressor

Part 3: Prediction based on monthly pollution levels (Jan–Dec).

```python
# Define features (monthly columns) and target
X = df.iloc[:, 4:16]  # Jan-Dec
y = df['2023']
```

[28] ✓ 0.0s Python

Split the data for test and train

```python
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

[29] ✓ 0.0s Python

Working with different models

```python
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)
lr_preds = lr_model.predict(X_test)
```

[30] ✓ 0.0s Python

```python
rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
rf_preds = rf_model.predict(X_test)
```

[31] ✓ 2.8s Python

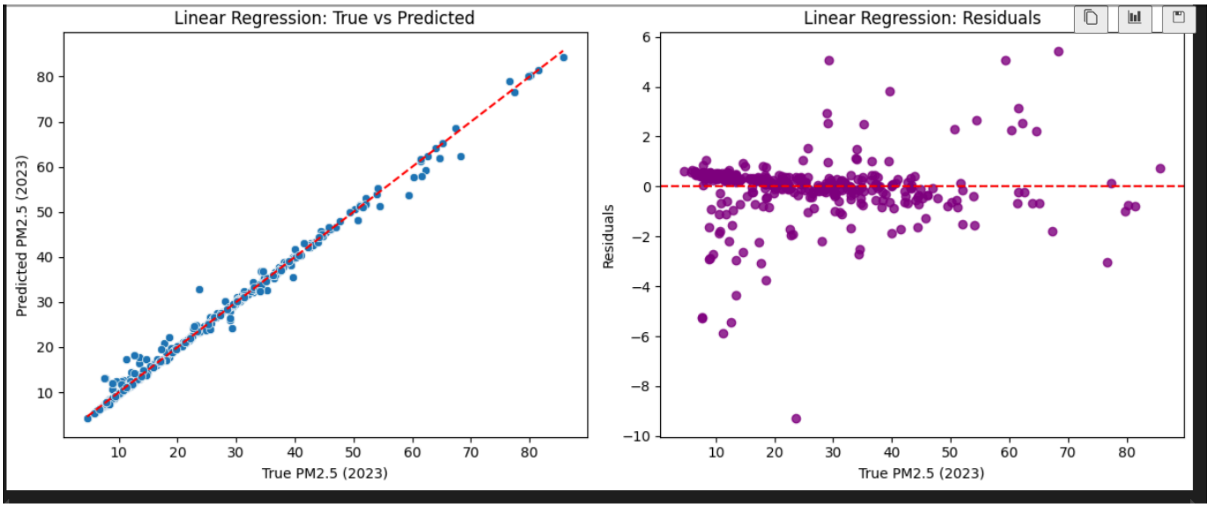
Results of Linear Regression and Random Forest models

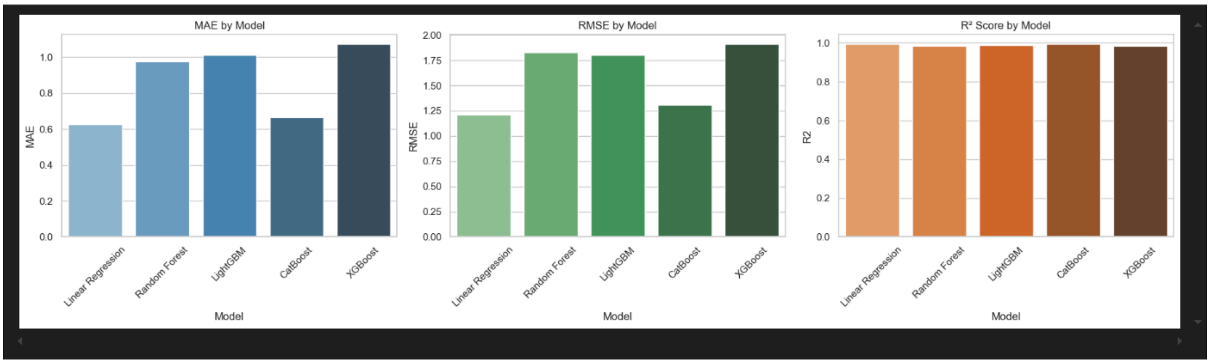
```python
def evaluate(true, preds, model_name):
    print(f"{model_name} Results:")
    print("MAE:", mean_absolute_error(true, preds))
    print("RMSE:", np.sqrt(mean_squared_error(true, preds)))
    print("R² Score:", r2_score(true, preds))
    print()

evaluate(y_test, lr_preds, "Linear Regression")
evaluate(y_test, rf_preds, "Random Forest")
```

[32] ✓ 0.0s Python

```
Linear Regression Results:
MAE: 0.625702445206461
RMSE: 1.2075254953581946
R² Score: 0.9939927807803642
```

Spaces: 4 Cell 24 of 46





✅ Conclusion

- **Best model**: **Linear Regression** — due to its lowest errors and highest R^2.
- **Runner-up**: **CatBoost** — excellent performance, especially if the data becomes more complex.
- **Other models** (LightGBM, XGBoost, RF): Still very good, but not better than the linear baseline here.

This shows that **sometimes simpler models outperform more complex ones**, especially when the data has a strong linear structure — which seems to be the case with monthly vs. annual PM2.5 levels.